Google **Web** Images Groups News Froogle New! **more »**

smart pointer (template or stl) Search Advanced Search Preferences

Lowercase "or" was ignored. Try "OR" to search for either of two terms. [details]

Web Results **1** - **10** of about **9,040** for **smart pointer (template or stl)**. (**0.44** seconds)

Smart Pointers - What, Why, Which?
... schemes using class-specific new and delete operators, **smart pointers** give you ... containers
and algorithms known as the standard **template** library (**STL**). ...
ootips.org/yonat/4dev/**smart-pointers**.html - 22k - Cached - Similar pages

An Overview of The Standard **Template** Library
... The **STL** provides class **templates** for maps, sets, multimaps ... Iterators are very similar
to **smart pointers** and have increment and dereferencing operations. ...
www.cs.rpi.edu/~wiseb/xrds/ovp2-3b.html - 26k - Cached - Similar pages

Using the The Standard **Template** Library with Borland C++
... an explicit call to a destructor via a **template** function. ... of the container is deleted,
the **STL** takes care ... Data>>; If you cannot use a **smart pointer**, then the ...
www.cs.rpi.edu/~wiseb/**stl**-borland.html - 26k - Cached - Similar pages
[More results from www.cs.rpi.edu]

Template std::auto_ptr< _Tp > class Reference
... operator=(). **template**<typename _Tp>. element_type& std::auto_ptr< _Tp >::operator
*, (,), const throw () [inline]. **Smart pointer** dereferencing. ...
gcc.gnu.org/onlinedocs/libstdc++/libstdc++-html-USERS-3.3/ classstd_1_1auto__ptr.html - 33k - Cached - Similar pages

Reference Counted **Smart Pointers**
... Reference Counted **Smart Pointers**. There is a problem with the auto_ptr **template**. ... These
surprise many programmers, and prevent its use in **STL** containers, as well ...
www.goingware.com/tips/parameters/refcount.html - 15k - Cached - Similar pages

flipCode Message Center - **Smart Pointers** and **STL** - General ...
... I know not all **smart pointers** work with **stl** ... And yes I have looked into Boost **pointers**,
and just ... int references; protected: string name; }; **template**< typename T ...
www.flipcode.com/ cgi-bin/msg.cgi?showThread=00005040&forum=general&id=-1 - 37k - Cached - Similar pages

flipCode Message Center - Re: COM **Smart Pointer** - Code Of The Day ...
... To contain CComPtrs inside an **STL** container, the CAdapt **template** is used ... said that,
Microsoft provides a better set of **smart pointer templates** which are ...
www.flipcode.com/ cgi-bin/msg.cgi?showThread=COTD-COMSmartPointer&forum=cotd&id=-1 - 52k - Cached - Similar pages
[More results from www.flipcode.com]

Links: C++
... used in a commercial **STL** implementation; Guru of the Week Archives - on a wide variety
of C++ topics; Memory Management **Templates** for C++ - **smart pointer** source; ...
www.technomagi.com/links/cpp.html - 13k - Cached - Similar pages

Mumit's **STL** Newbie guide
... This is where **Smart pointers** shine. ... the storage required to hold the **pointer**, not

the ... templated deleters like the following: **template** <class ForwardIterator ...
www.xraylith.wisc.edu/~khan/software/ **stl**/STL.newbie.html - 101k - Cached - Similar pages

STL-friendly **smart pointer**? - GameDev.Net Forums
... I'm looking into **smart pointers** to help memory management in my ... problems when used with the **STL**, such as ... For one thing, partial **template** specialization isn't ...
www.gamedev.net/community/forums/ topic.asp?topic_id=198201 - 25k - Cached - Similar pages

Goooooooooogle ►
Result Page: 1 2 3 4 5 6 7 8 9 10 **Next**

smart pointer (template or stl) Search

Search within results | Language Tools | Search Tips | Dissatisfied? Help us improve

Google Home - Advertising Solutions - Business Solutions - About Google

Google

Web Images Groups News Froogle New! **more »**

(safe or smart) pointer template check null | Search | Advanced Search | Preferences

Lowercase "or" was ignored. Try "OR" to search for either of two terms. [details]

Web Results **11** - **20** of about **6,260** for **(safe or smart) pointer template check null**. (**0.22** seconds)

Smart Pointers
... in C and C++ Part 3: **Safe** Programming with ... Consequently, any use of **Smart**-Ptr's default
constructor yields ... which does not accept **null pointers** as initializers ...

www.informit.com/ articles/article.asp?p=25264&seqNum=15 - 32k - Cached - Similar pages
[More results from www.informit.com]

Otaku, Cedric's weblog: Comment on Groovy's -> and . operators
... that you do not need to **check** every method ... These can provide exception-**safe** heap management,
or could ... So say I have a **smart-pointer template** class **Pointer** that ...

www.beust.com/weblog/mt-cedricqwerty.cgi?entry_id=103 - 9k - Cached - Similar pages

Semple Analysis Engine: semple::Node class Reference
... These routines provide **safe** downcasting and derived type **checking** for ... a class derived from T; otherwise, returns the **null (smart) pointer**. ... **template**<typename T>. ...
www.cs.rpi.edu/~gregod/Semple/classsemple_1_1Node.html - 19k - Cached - Similar pages

Introduction I've recently been dealing with one of those ...
... on the heap and managed by a reference-counted **smart pointer**. ... delete its target object, it is **safe** to do ... **pointer** serves two purposes (flag and **pointer** to target ...
home.clara.net/raoulgough/weak_ptr/weak_ptr.txt - 17k - Cached - Similar pages

CodeGuru: **Smart Pointer** (with Object Level Thread Synchronization ...
... and ACCESS is the class used to get thread **safe** access to ... SmartPtr class has a non **template** base class SmartPtrBase, which is made ... **Smart Pointer** (with Object ...
www.codeguru.com/Cpp/Cpp/cpp_mfc/ **pointers**/article.php/c829/ - 48k - Cached - Similar pages

flipCode Message Center - Re: COM **Smart Pointer** - Code Of The Day ...
... a problem, but better be on the **safe** side. ... html Use a trait / policy based design in your **smart pointer**. In code terms, add a second **template** parameter that is ...
www.flipcode.com/ cgi-bin/msg.cgi?showThread=COTD-COMSmartPointer&forum=cotd&id=-1 - 52k - Cached - Similar pages
[More results from www.flipcode.com]

MMAccessor.h Source File
... behaves as a 00040 * '**smart pointer**', which overloads ... 00047 */ 00048 **template**<class MC_t> 00049 class ... member fields, which should be **safe**) */ 00055 MMAccessor ...
www-2.cs.cmu.edu/~tekkotsu/dox/ MMAccessor_8h-source.html - 23k - Cached - Similar pages

[PDF] Refinements to **Smart Pointers**
File Format: PDF/Adobe Acrobat - View as HTML
... If type-**safe** casts (ie, dynamic_casts) are to be ... the latest rules governing **template**

instantiation and how ... Memory Management and **Smart Pointers**," C++ Report ...
www.aristeia.com/Papers/C++ReportColumns/novdec96.pdf - Similar pages

Smart & Weak **Pointers** - valuable tools for games Charles Bloom, 3 ...
... Now, you make a SmartPtr<> **template** that automatically takes a ... the end of the function, which keeps you **safe**. ... You can also hold temporary **smart pointers** if you ...
www.cbloom.com/3d/techdocs/**smart_pointers**.txt - 8k - Cached - Similar pages

TemplatenRef< TYPE > class Reference
... nRef, nAutoRef, and nDynAutoRef implement **safe pointers** to objects. ... it's a class **template** implementing a **smart pointer**. ... of this refernce as a nRoot **pointer**. ...
nebuladevice.sourceforge.net/doc/doxydoc/ nebulacore/html/classnRef.html - 27k - Cached - Similar pages

[boost] Re: **Smart Pointer** Timing Test Results :: ASPN Mail Archive ...
... static_cast<element_type*> (ptr); // This is **safe** since there is ... vptr_rc { void * ptr; long count; }; **template** <class T ... as well as another **smart pointer** to the ...
aspn.activestate.com/ASPN/Mail/Message/boost/1139090 - 48k - Cached - Similar pages

◀ Goooooooooooogle ▶
Result Page: **Previous** 1 2 3 4 5 6 7 8 9 10 11 **Next**

(safe or smart) pointer template check null | Search

Search within results | Language Tools | Search Tips

Google Home - Advertising Solutions - Business Solutions - About Google

	Last Name	First Name
Search Another: Inventor	APPLIN	JOHN

Search

To go back use Back button on your browser toolbar.

Back to PALM | ASSIGNMENT | OASIS | Home page

PORTAL
US Patent & Trademark Office

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: ● The ACM Digital Library ○ The Guide

"safe pointer"

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used **safe pointer** Found **27** of **132,857**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

□ Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 27 Result page: **1** 2 next

Relevance scale

1 Safe pointers
Christoph Grein
December 1999 **ACM SIGAda Ada Letters**, Volume XIX Issue 4
Full text available: pdf(229.30 KB) Additional Information: full citation, index terms

2 Efficient detection of all pointer and array access errors
Todd M. Austin, Scott E. Breach, Gurindar S. Sohi
June 1994 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1994 conference on Programming language design and implementation**, Volume 29 Issue 6
Full text available: pdf(1.62 MB) Additional Information: full citation, abstract, references, citings, index terms

We present a pointer and array access checking technique that provides complete error coverage through a simple set of program transformations. Our technique, based on an extended safe pointer representation, has a number of novel aspects. Foremost, it is the first technique that detects all spatial and temporal access errors. Its use is not limited by the expressiveness of the language; that is, it can be applied successfully to compiled or interpreted languages with subscripted and mutabl ...

3 CCured: type-safe retrofitting of legacy code
George C. Necula, Scott McPeak, Westley Weimer
January 2002 **ACM SIGPLAN Notices , Proceedings of the 29th ACM SIGPLAN-SIGACT symposium on Principles of programming languages**, Volume 37 Issue 1
Full text available: pdf(1.65 MB) Additional Information: full citation, abstract, references, citings

In this paper we propose a scheme that combines type inference and run-time checking to make existing C programs type safe. We describe the CCured type system, which extends that of C by separating pointer types according to their usage. This type system allows both pointers whose usage can be verified statically to be type safe, and pointers whose safety must be checked at run time. We prove a type soundness result and then we present a surprisingly simple type inference algorithm that is able ...

4 CCured in the real world
Jeremy Condit, Matthew Harren, Scott McPeak, George C. Necula, Westley Weimer
May 2003 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2003 conference on Programming language design and implementation**, Volume 38 Issue 5

Full text available: pdf(262.59 KB) Additional Information: full citation, abstract, references, index terms

CCured is a program transformation system that adds memory safety guarantees to C programs by verifying statically that memory errors cannot occur and by inserting run-time checks where static verification is insufficient.This paper addresses major usability issues in a previous version of CCured, in which many type casts required the use of pointers whose representation was expensive and incompatible with precompiled libraries. We have extended the CCured type inference algorithm to recognize a ...

Keywords: C, compatibility with library code, memory safety, run-time type information, type safety

5 Arrays and references
Jean-Jacques Girardot
May 1990 **ACM SIGAPL APL Quote Quad , Conference proceedings on APL 90: for the future**, Volume 20 Issue 4
Full text available: pdf(1.21 MB) Additional Information: full citation, abstract, references, citings, index terms

Generalized arrays in APL have been for long a very controversial subject. Since we are now undertaking the redaction of an extended standard for APL, it seems legitimate to reopen the old debate. An analysis of both nested and boxed array systems, in the light of a new development in APL which consists of the introduction of a new data-type in the language, shows the interest of having both systems with their own specificities.

6 Protecting C programs from attacks via invalid pointer dereferences
Suan Hsi Yong, Susan Horwitz
September 2003 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 9th European software engineering conference held jointly with 10th ACM SIGSOFT international symposium on Foundations of software engineering**, Volume 28 Issue 5
Full text available: pdf(526.15 KB) Additional Information: full citation, abstract, references, index terms

Writes via unchecked pointer dereferences rank high among vulnerabilities most often exploited by malicious code. The most common attacks use an unchecked string copy to cause a buffer overrun, thereby overwriting the return address in the function's activation record. Then, when the function "returns", control is actually transferred to the attacker's code. Other attacks may overwrite function pointers, setjmp buffers, system-call arguments, or simply corrupt data to cause a denial of service.A ...

Keywords: buffer overrun, instrumentation, security, static analysis

7 Triangulations in CGAL (extended abstract)
Jean-Daniel Boissonnat, Olivier Devillers, Monique Teillaud, Mariette Yvinec
May 2000 **Proceedings of the sixteenth annual symposium on Computational geometry**
Full text available: pdf(793.02 KB) Additional Information: full citation, references, citings, index terms

8 BlackBox: a new object-oriented framework for CS1/CS2
J. Stanley Warford
March 1999 **ACM SIGCSE Bulletin , The proceedings of the thirtieth SIGCSE technical symposium on Computer science education**, Volume 31 Issue 1
Full text available: Additional Information: full citation, abstract, references, citings, index

pdf(645.61 KB) terms

This paper describes the BlackBox framework, an object-oriented application development environment, and our experience with its use over the past several years in the CS1/CS2 course. This little-known framework features: (1) a graphical user interface that is simple enough for beginning students to program, (2) true cross-platform capability, (3) guaranteed memory-safe pointers with automatic garbage collection, (4) a new language, Component Pascal, that combines the best of Java and Pascal, (5 ...

Keywords: BlackBox, CS1, CS2, component Pascal, formal methods, frameworks, programming languages

9 Using Ada and C++ in computer science education
Raymond J. Toal
January 1996 **ACM SIGAda Ada Letters**, Volume XVI Issue 1
Full text available: pdf(754.31 KB) Additional Information: full citation, abstract, index terms

Undergraduate students of Computer Science or Software Engineering must become familiar with imperative programming languages, due to the extensive use of these languages in industry. Perhaps the two most interesting imperative languages, from a technical standpoint, are Ada and C++, as these two languages include a number of modern features and enjoy widespread popularity. We argue that a four-year undergraduate curriculum in Computer Science which emphasizes imperative programming languages be ...

10 LLVA: A Low-level Virtual Instruction Set Architecture
Vikram Adve, Chris Lattner, Michael Brukman, Anand Shukla, Brian Gaeke
December 2003 **Proceedings of the 36th Annual IEEE/ACM International Symposium on Microarchitecture**
Full text available: pdf(196.08 KB) Publisher Site Additional Information: full citation, abstract

A virtual instruction set architecture (V-ISA) implementedvia a processor-specific software translation layercan provide great flexibility to processor designers. Recentexamples such as Crusoe and DAISY, however, haveused existing hardware instruction sets as virtual ISAs,which complicates translation and optimization. In fact,there has been little research on specific designs for a virtualISA for processors. This paper proposes a novel virtualISA (LLVA) and a translation strategy for implementi ...

11 A truly generative semantics-directed compiler generator
Harald Ganzinger, Robert Giegerich, Ulrich Möncke, Reinhard Wilhelm
June 1982 **ACM SIGPLAN Notices , Proceedings of the 1982 SIGPLAN symposium on Compiler construction**, Volume 17 Issue 6
Full text available: pdf(918.86 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper describes semantic processing in the compiler generating system MUG2. MUG2 accepts high-level descriptions of the semantics of a programming language including full runtime semantics, data flow analysis, and optimizing transformations. This distinguishes MUG2 from systems such as YACC [Joh75], HLP [HLP78], PQCC [PQC79], or its own former version [GRW77] with respect to expressive power and convenience. In this respect, MUG2 comes close to semantics-directed systems such as [Mos76 ...

12 How to write system-specific, static checkers in metal
Benjamin Chelf, Dawson Engler, Seth Hallem
November 2002 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 2002 ACM SIGPLAN-SIGSOFT workshop on Program analysis for software**

tools and engineering, Volume 28 Issue 1
Full text available: pdf(190.85 KB) Additional Information: full citation, references, index terms

13 Session: Modula-3 network objects over ANSA: heterogeneous object-based RPC in a modern systems programming language
David Evers, Peter Robinson
September 1992 **Proceedings of the 5th workshop on ACM SIGOPS European workshop: Models and paradigms for distributed systems structuring**
Full text available: pdf(453.29 KB) Additional Information: full citation, abstract, references

Modula-3 provides language-level features such as threads, objects and exceptions which are useful in distributed systems. The ANSA testbench provides a complete infrastructure for object-based distributed systems, but currently requires the use of C as the main programming language. We describe a successful attempt to marry the two, which provides a practical example of how a modern systems programming language can make the construction of object-based distributed systems more congenial for the ...

14 Automated discovery of scoped memory regions for real-time Java
Morgan Deters, Ron K. Cytron
June 2002 **ACM SIGPLAN Notices , Proceedings of the third international symposium on Memory management**, Volume 38 Issue 2 supplement
Full text available: pdf(227.49 KB) Additional Information: full citation, abstract, references, citings, index terms

Advances in operating systems and languages have brought the ideal of reasonably-bounded execution time closer to developers who need such assurances for real-time and embedded systems applications. Recently, extensions to the Java libraries and virtual machine have been proposed in an emerging standard, which provides for specification of release times, execution costs, and deadlines for a restricted class of threads. To use such features, the code executing in the thread must never reference s ...

Keywords: garbage collection, memory management, real-time Java, regions, trace-based analysis

15 The design and implementation of panar maps in CGAL
Eyal Flato, Dan Halperin, Iddo Hanniel, Oren Nechushtan, Eti Ezra
December 2000 **Journal of Experimental Algorithmics (JEA)**, Volume 5
Full text available: pdf(934.22 KB) ps(3.78 MB) LaTeX(18.00 bytes) Additional Information: full citation, citings

16 Bugs as deviant behavior: a general approach to inferring errors in systems code
Dawson Engler, David Yu Chen, Seth Hallem, Andy Chou, Benjamin Chelf
October 2001 **ACM SIGOPS Operating Systems Review , Proceedings of the eighteenth ACM symposium on Operating systems principles**, Volume 35 Issue 5
Full text available: pdf(1.53 MB) Additional Information: full citation, abstract, references, citings, index terms

A major obstacle to finding program errors in a real system is knowing what correctness rules the system must obey. These rules are often undocumented or specified in an ad hoc manner. This paper demonstrates techniques that automatically extract such checking information from the source code itself, rather than the programmer, thereby avoiding the need for a priori knowledge of system rules.The cornerstone of our approach is inferring programmer "beliefs" that we then cross-check for contradict ...

17 Design and implementation of generics for the .NET Common language runtime
Andrew Kennedy, Don Syme
May 2001 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2001 conference on Programming language design and implementation**, Volume 36 Issue 5
Full text available: pdf(1.25 MB) Additional Information: full citation, abstract, references, citings, index terms

The Microsoft.NET Common Language Runtime provides a shared type system, intermediate language and dynamic execution environment for the implementation and inter-operation of multiple source languages. In this paper we extend it with direct support for parametric polymorphism (also known as generics), describing the design through examples written in an extended version of the C# programming language, and explaining aspects of implementation by reference to a prototype extension to the runtim ...

18 Checkmate: cornering C++ dynamic memory errors with checked pointers
Scott M. Pike, Bruce W. Weide, Joseph E. Hollingsworth
March 2000 **ACM SIGCSE Bulletin , Proceedings of the thirty-first SIGCSE technical symposium on Computer science education**, Volume 32 Issue 1
Full text available: pdf(554.01 KB) Additional Information: full citation, abstract, references, index terms

Pointer errors are stumbling blocks for student and veteran programmers alike. Although languages such as Java use references to protect programmers from pointer pitfalls, the use of garbage collection dictates that languages like C++ will still be used for real-time mission-critical applications. Pointers will stay in the classroom as long as they're used in industry, so as educators, we must find better ways to teach them. This paper presents checked pointers, a simple wr ...

19 Techniques for trusted software engineering
Premkumar T. Devanbu, Philip W-L Fong, Stuart G. Stubblebine
April 1998 **Proceedings of the 20th international conference on Software engineering**
Full text available: pdf(1.21 MB) Publisher Site Additional Information: full citation, references, citings, index terms

20 Pointer analysis for programs with structures and casting
Suan Hsi Yong, Susan Horwitz, Thomas Reps
May 1999 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1999 conference on Programming language design and implementation**, Volume 34 Issue 5
Full text available: pdf(1.65 MB) Additional Information: full citation, abstract, references, citings, index terms

Type casting allows a program to access an object as if it had a type different from its declared type. This complicates the design of a pointer-analysis algorithm that treats structure fields as separate objects; therefore, some previous pointer-analysis algorithms "collapse" a structure into a single variable. The disadvantage of this approach is that it can lead to very imprecise points-to information. Other algorithms treat each field as a separate object based on its offset and size. While ...

Results 1 - 20 of 27 Result page: **1** 2 next

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Player